CANNY EDGE DETECTOR

General problem

There is an unavoidable trade-off between noise reduction and edge localization

Canny is found to be optimal for step edges corrupted by white noise

Canny exploits three criteria covering:

- Detection (no missing edges, no spurious edges)
- Localization (minimal distance between true and detected edge position)
- One response (minimize multiple responses to single edges)

Localize edges by non-maximal suppression

Thin the wide ridges around local maxima in gradient magnitude down to edges that are one pixel wide.

Assuming we have already computed the gradient magnitude and direction stored in image g and θ, respectively, and g_s is the output, the algorithm becomes:

$$
\text{for all (x,y)} \quad \begin{cases} \text{Approximate} \quad \theta(x,y) \quad \text{by} \quad \hat{\theta} \in \{0, \pi/4, \pi/2, 3\pi/4\} \\ \text{if } \; g(x,y) < g \text{ at neighbour in direction } \hat{\theta} \text{ or } \hat{\theta}+\pi \\ \qquad g_s(x,y) = 0 \\ \text{else} \\ \qquad g_s(x,y) = g(x,y) \end{cases}
$$

Eliminate spurious edges by hysteresis thresholding

1- Use two thresholds T_{low} and T_{high}

2- Use T_{high} for selecting the best edge pixel candidates

3- Grow these pixels into contours by searching for neighbours with gradient magnitudes higher than T_{low}

Reduces the number of false positives because edges are tracked only if at least one pixel has a gradient exceeding T_{high}

Reduces fragmentation of contours by allowing significant fluctuations in gradient magnitude if T_{low} is suitable small

Canny edge detector algorithm

1- Convolve the image with a Gaussian of scale sigma

2- Compute gradient magnitude and direction

3- Localize edges by non-maximal suppression

4- Eliminate spurious edges by hysteresis thresholding

(5)- Repeat steps 1-4 for ascending values of sigma and aggregate the final edge information. Step 5 rarely implemented in practice.